

A  
Major Project  
On  
**HONEYPOT FOR CLOUD SECURITY**

(Submitted in partial fulfillment of the requirements for the award of Degree)

**BACHELOR OF TECHNOLOGY**

in  
**INFORMATION TECHNOLOGY**

**BY**

Mohammed Zubair Hussain (177R1A1238)

J. Venkat. P. S. V. V (177R1A1220)

CH. Naveen (177R1A1204)

Under the Guidance of  
**Dr. B. KAVITHA RANI**  
(Professor)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New  
Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2017 -2021

## **DEPARTMENT OF INFORMATION TECHNOLOGY**



### **CERTIFICATE**

This is to certify that the project entitled “**HONEYPOT FOR CLOUD SECURITY**” being submitted by **MOHAMMED ZUBAIR HUSSAIN (177R1A1238), J. VENKAT. P. S. V. V (177R1A1220) & CH. NAVEEN (177R1A1204)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Information Technology of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2020-2021.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**INTERNAL GUIDE**  
**Dr. B. Kavitha Rani**  
**(Professor)**

**DIRECTOR**  
**Dr. A. Raji Reddy**

**HOD**

**EXTERNAL EXAMINER**

**Submitted for viva voce Examination held on \_\_\_\_\_**

## ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express our profound gratitude and deep regard to our guide **Dr. B. Kavitha Rani**, for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane and Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the **Head of the Department** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project.

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

**MOHAMMED ZUBAIR HUSSAIN (177R1A1238)**

**J. VENKAT. P. S.V.V (177R1A1220)**

**CH. NAVEEN (177R1A1204)**

## **ABSTRACT**

The World is getting Smaller thanks to technology. With the rapid increase in the number of users, there is a rise in issues related to hardware failure, web hosting, space and memory allocation of data, which is directly or indirectly leading to the loss of data. With the objective of providing services that are reliable, fast and low in cost, we turn to cloud-computing practices. With a tremendous development in this technology, there is ever increasing chance of its security being compromised by malicious users. A way to divert malicious traffic away from systems is by using Honeypot. The advantages of a low- interaction honeypot is their simplicity.

These honeypots tend to be easier to deploy and maintain, with minimal risk. Usually they involve installing software, selecting the operating systems and services you want to emulate and monitor, and letting the honeypot go from there. It is a colossal strategy that has shown signs of improvement in security of systems. Keeping in mind the various legal issues one may face while deploying Honeypot on third-party cloud vendor servers, the concept of Honeypot is implemented in a file-sharing application which is deployed on cloud server. This discusses the detection attacks in a cloud-based environment as well as the use of Honeypot for its security, thereby proposing a new technique to do the same.

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
Fig. 3.1	Project Architecture	6
Fig. 3.2	Use case diagram	8
Fig. 3.3	Class diagram	9
Fig. 3.4	Sequence diagram	10
Fig. 3.5	Activity diagram	11

## **LIST OF SCREENSHOTS**

<b>SCREENSHOT NO.</b>	<b>SCREENSHOT NAME</b>	<b>PAGE NO.</b>
5.1. Screenshot	Home Page	17
5.1.1. Screenshot	Home Page Menu	17
5.2. Screenshot	Login Page	18
5.3. Screenshot	Admin Login Menu	18
5.3.1. Screenshot	User Details	19
5.3.2. Screenshot	Acceptance Page for Files	19
5.3.3. Screenshot	List of Files Shared by Users	20
5.4. Screenshot	Registration Form	20
5.5. Screenshot	User Login Menu	21
5.5.1. Screenshot	User Sends File	21
5.5.2. Screenshot	Files Shared by Users	22
5.5.3. Screenshot	Files Received by Users	22
5.6. Screenshot	Attacker Login Form	23
5.6.1. Screenshot	Attacker Login Menu	23

# TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURES	ii
LIST OF SCREENSHOTS	iii
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 PROJECT SCOPE	1
1.2 PROJECT PURPOSE	1
1.3 PROJECT FEATURES	1
<b>2. SYSTEM ANALYSIS</b>	<b>2</b>
2.1 PROBLEM DEFINITION	2
2.2 EXISTING SYSTEM	2
2.2.1 LIMITATION OF EXISTING SYSTEM	3
2.3 PROPOSED SYSTEM	3
2.3.1 ADVANTAGES OF PROPOSED SYSTEM	3
2.4 FEASIBILITY STUDY	4
2.4.1 ECONOMIC FEASIBILITY	4
2.4.2 TECHNICAL FEASIBILITY	4
2.4.3 BEHAVIORAL FEASIBILITY	5
2.5 HARDWARE & SOFTWARE REQUIREMENTS	5
2.5.1 HARDWARE REQUIREMENTS	5
2.5.2 SOFTWARE REQUIREMENTS	5
<b>3. ARCHITECTURE</b>	<b>6</b>
3.1 PROJECT ARCHITECTURE	6
3.2 MODULES DESCRIPTION	6
3.2.1 USER	6
3.2.2 ADMIN	7
3.2.3 ATTACKER	7

3.3 USE CASE DIAGRAM	8
3.4 CLASS DIAGRAM	9
3.5 SEQUENCE DIAGRAM	10
3.6 ACTIVITY DIAGRAM	11
<b>4. IMPLEMENTATION</b>	<b>12</b>
4.1 SAMPLE CODE	12
<b>5. SCREEN SHOTS</b>	<b>17</b>
<b>6. TESTING</b>	<b>24</b>
6.1 INTRODUCTION TO TESTING	24
6.2 TYPES OF TESTING	24
6.2.1 UNIT TESTING	24
6.2.2 INTEGRATION TESTING	24
6.2.3 FUNCTIONAL TESTING	25
6.3 TEST CASES	25
6.3.1 TEST CASES PASSED	25
6.3.2 TEST CASES FAILED	26
<b>7. CONCLUSION</b>	<b>27</b>
7.1 PROJECT CONCLUSION	27
7.2 FUTURE ENHANCEMENT	27
<b>8. BIBLIOGRAPHY</b>	<b>28</b>
8.1 GITHUB LINK	28
8.2 REFERENCES	28
8.3 WEBSITES	28



# **1.INTRODUCTION**

## **1. INTRODUCTION**

### **1.1 PROJECT SCOPE**

Cloud-based Honeypots give the capacity to explore and examine assaults that hit ordinary customers. Having them permits a specialist to interpret the IP locations and malware being utilized into security content that can ensure a normal cloud environment. Once those IP addresses have been distinguished, they will then lead a ping scope and defenselessness output to discover a shortcoming in the system outline or vulnerabilities in programming that can be misused.

### **1.2 PROJECT PURPOSE**

Cloud computing is a technique to store, share and access data anytime and anywhere with a device that is connected to a network, preferably the internet. Honeypots are viewed as a successful technique to track programmer conduct and uplift the viability of security instruments. Honeypots are specifically designed to not only purposely engage and deceive hackers but also identify malicious activities performed over the Internet and can be counted as an effective method to track hacker behavior. Their aim is to analyze, understand, watch and track attacker's behavior in order to create systems that are not only secure but can also handle such traffic.

### **1.3 PROJECT FEATURES**

The main features of this project are that the designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

## **2. SYSTEM ANALYSIS**

## **2. SYSTEM ANALYSIS**

### **SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

#### **2.1 PROBLEM DEFINITION**

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

#### **2.2 EXISTING SYSTEM**

Cloud computing consists of an expandable storage space with no physical storage space which is accessible from anywhere in the world using any device, by connecting it to the internet. Honeypots can be defined as systems or assets which are used to not only trap, monitor but to also identify erroneous requests present within a

network. In existing system Cloud-based Honeypots give the capacity to explore and examine assaults that hit ordinary customers, they will then lead a ping scope and defenselessness output to discover a shortcoming in the system outline or vulnerabilities in programming that can be misused.

### **2.2.1 LIMITATIONS OF EXISTING SYSTEM**

- Existing works are conspicuous yet genuine.
- Mostly concentration of the cloud security is given to the weakest node.
- Existing work might not be able to differentiate genuine user requests and erroneous requests

### **2.3 PROPOSED SYSTEM**

This project proposes a new technique of protecting data and resources in a cloud through Honeypot by implementing it through an application on the above-mentioned infrastructure (Cloud Computing Environment). There are many restraints that need to be followed while implementing a Honeypot. The application makes it possible to store as well as share a document. While sharing or uploading the document it is encrypted using a password. If the correct password is not given then no message would be displayed rather the attacker would be shown an empty file. Since the actual working of a Honeypot involves silent detection, hence the application tracks the IP address of the user so that later the admin can review it and recognize the malicious entity.

#### **2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

- The actual working of a Honeypot involves silent detection.
- IP Address Tracking.
- Silently detects the Hacker.
- Doesn't let the hacker know that he/she has been identified as attacker.

## **2.4 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

### **2.4.1 ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

### **2.4.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

## 2.5 HARDWARE & SOFTWARE REQUIREMENTS

### 2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel i3 or above
- Storage : 500GB or more
- RAM : 8GB or more

### 2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating System : Windows 8.1, 10/ Linux
- User Interface : HTML, CSS
- Client-side Scripting : JavaScript
- Programming Language : Java
- Web Applications : JDBS, Servlets, JSP
- IDE/Workbench : My Eclipse
- Database : MySQL
- Server Deployment : Tomcat 9.0

# **3. ARCHITECTURE**



### 3. ARCHITECTURE

#### 3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for breed detection using machine learning, starting from input to final prediction.

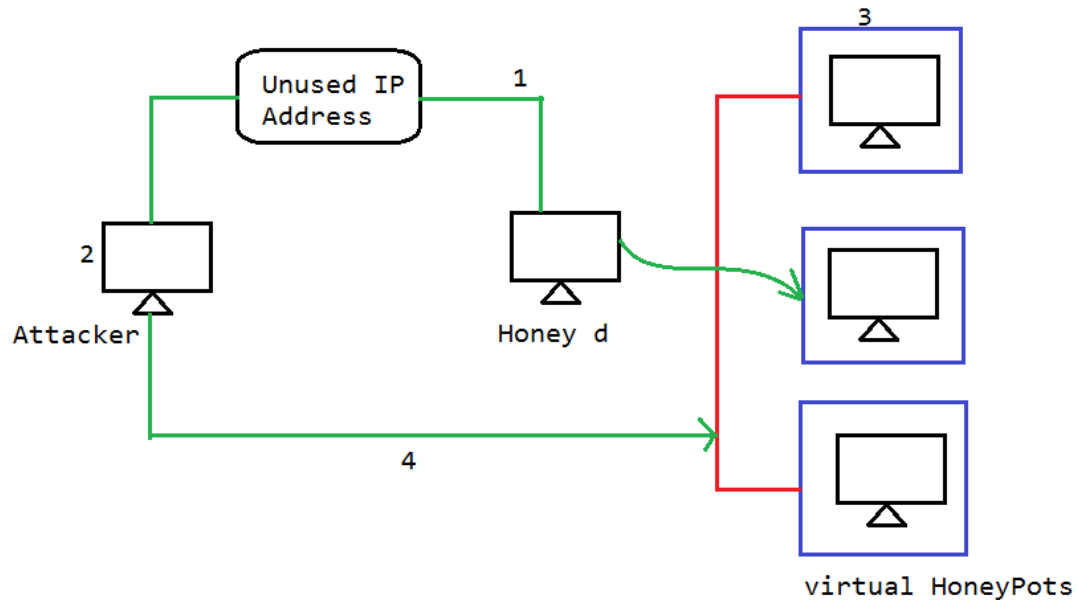


Fig 3.1: Project Architecture of Honeypot for Cloud Security

#### 3.2 MODULES DESCRIPTION

##### Modules

- **User**
- **Admin**
- **Attacker**

##### 3.2.1 User

- User need to Register with his/her details
- After User's network created by router, user can login with email id and password
- Can select any files, documents, images etc.
- Upload it to their own cloud space.
- Share the uploaded files to other users.

- Download their uploaded file and received files.
- Logout as their work is done.

### **3.2.2 Admin**

- Admin will login with his/her credentials.
- They can create user network by accepting the network request.
- All the users IP addresses are visible to the admin/router.
- All the files shared among the users are visible to the admin
- Admin has to verify and accept the request of file being shared among the users.
- Can view all the feedbacks from the users.
- Logout as work is done.

### **3.2.3 Attacker**

- Attacker will login with one of the user's credentials.
- Can choose any of the options, with/without honeypot.
- If Without honeypot is chosen then file's data is visible to the attacker.
- If with honeypot is chosen then file's data is invisible to the attacker.

### 3.3 USE CASE DIAGRAM

In the use case diagram, we have basically three actors who are the user, attacker, and the admin/router. The user has the right to login, send, receive, download, view files. The attacker can log in and chose any of the options of with/without Honeypot. The admin can create a network for new users, view IP Addresses, accept or share files, and view feedback from the users.

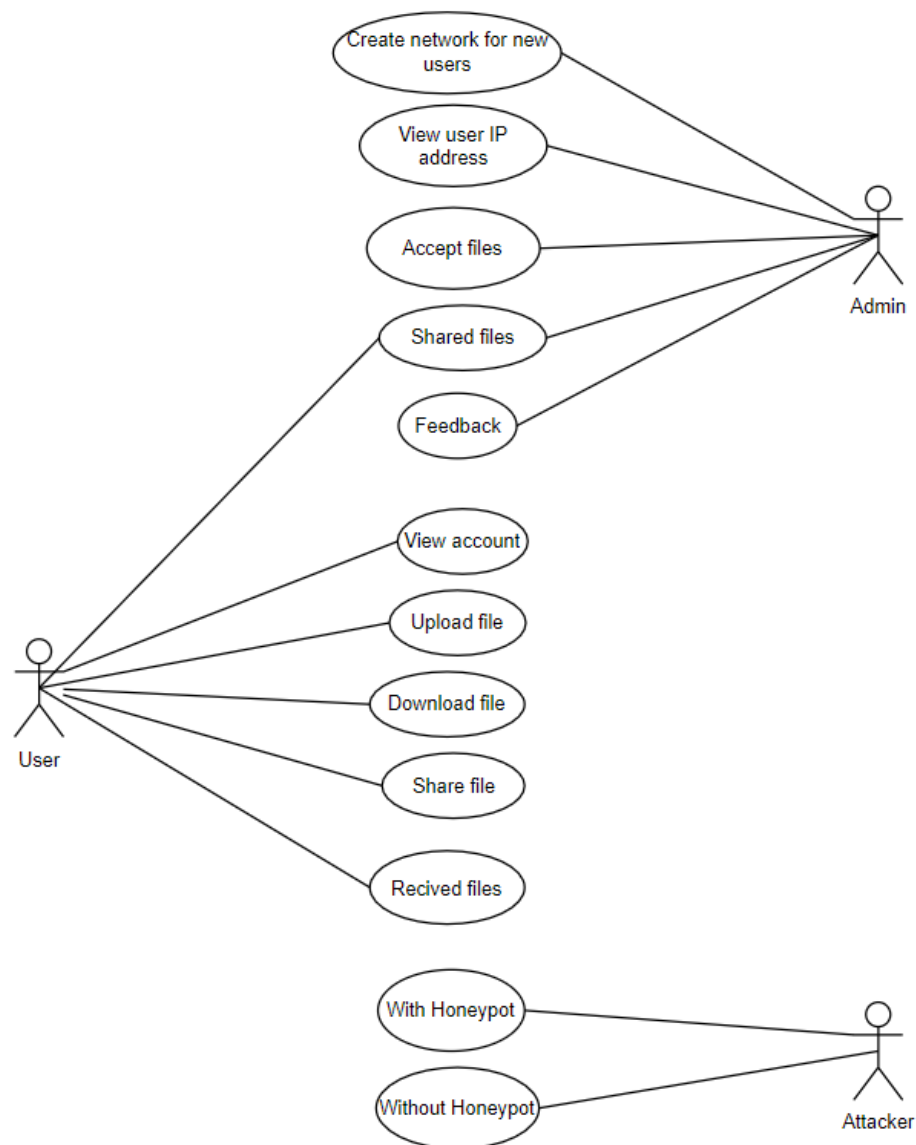


Fig 3.2: Use Case Diagram of Honeypot for Cloud Security

### 3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects. A class diagram in UML is a type of static structure diagram that describes the structure of a system by showing the system classes, their attributes, operations and relationships among objects.

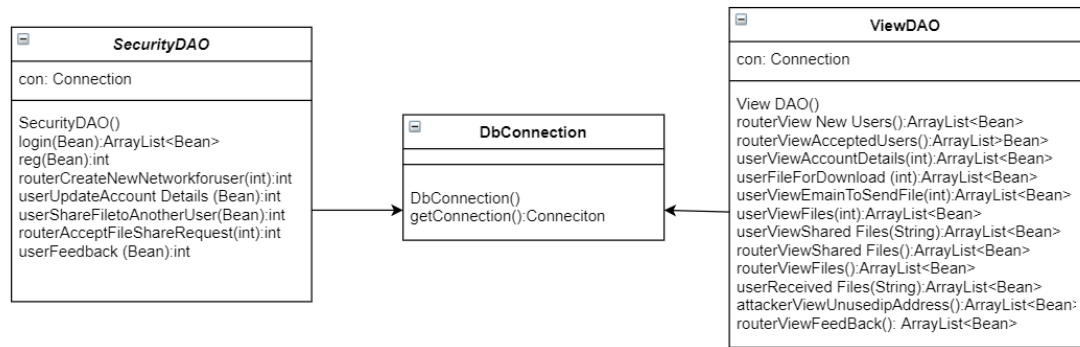


Fig 3.3: Class Diagram for Honeypot for Cloud Security

### 3.5 SEQUENCE DIAGRAM

The user logs in after the admin has created the network and selects the files and send to the respective user. The attacker logs in and tries to view the files but Honeypot mechanism hides the content of the data. A sequence diagram simply depicts interaction between objects in a sequential order that is the order in which these interactions take place.

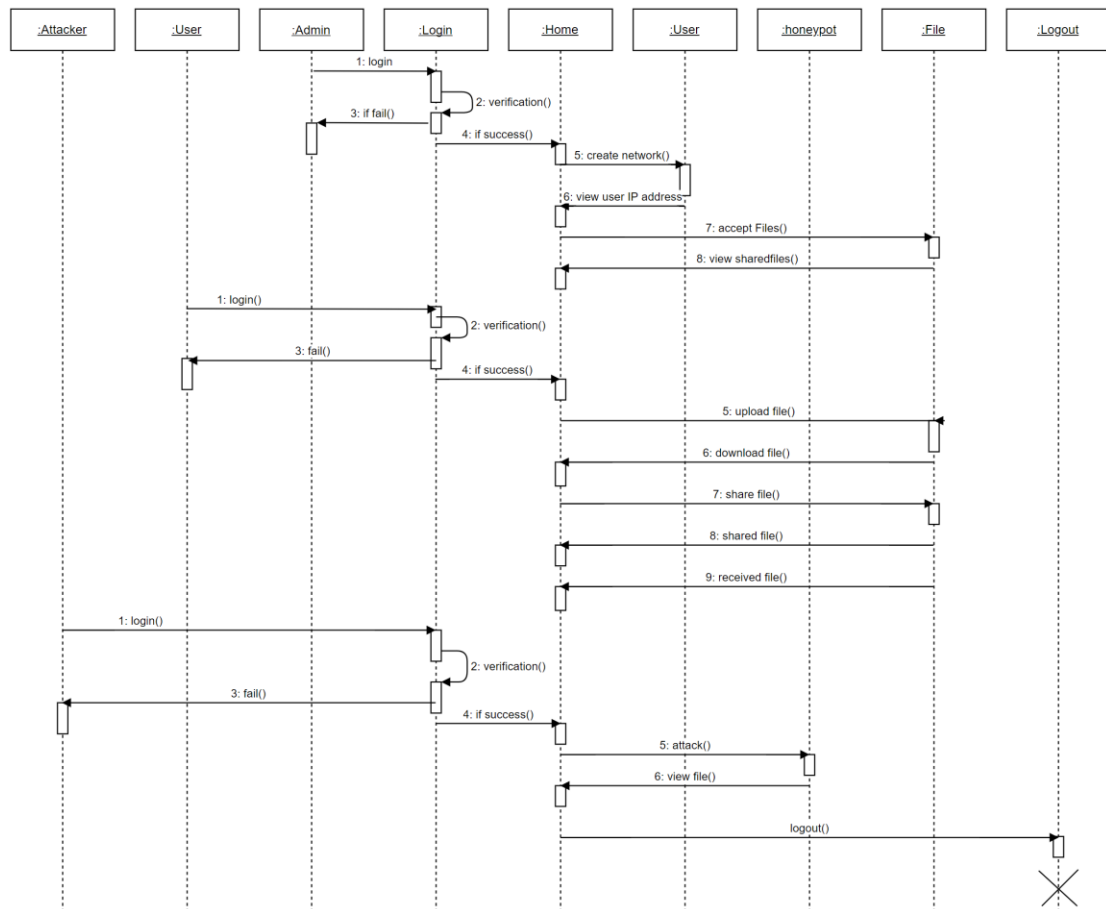


Fig 3.4: Sequence Diagram for Honeypot for Cloud Security

### 3.6 ACTIVITY DIAGRAM

It describes about flow of activity states. Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the systems. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

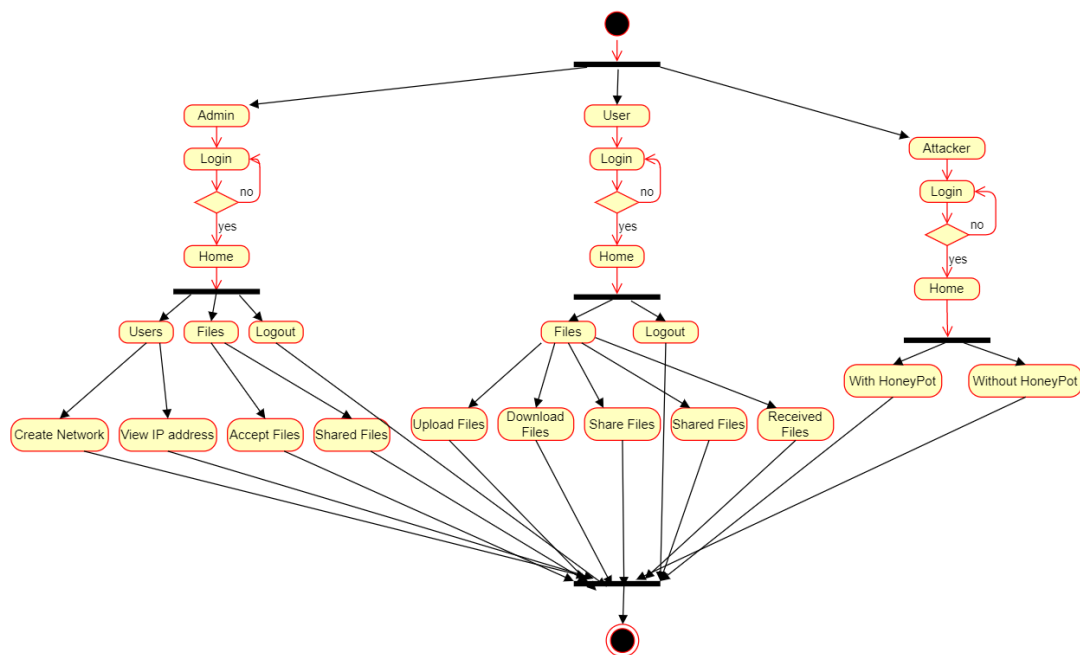


Fig 3.5: Activity Diagram for User for Honeypot for Cloud Security

## **4. IMPLEMENTATION**

## 4. IMPLEMENTATION

### 4.1 SAMPLE CODE

#### **Bean.java**

```

package com.honeypot.project.bean;
import java.sql.Blob;
public class Bean {
    private String uname;
    private String password;
    private String email;
    private String mobile;
    private String date;
    private String utype;
    private int uid;
    private String card;
    private String address;
    private String ipaddress;
    private byte[] file ;
    private String fname;
    private int fid;
    private Blob blob;
    private String status;

    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public Blob getBlob() {
        return blob;
    }
    }1
    public void setBlob(Blob blob) {

```



```
        this.blob = blob;
    }
    public int getFid() {
        return fid;
    }
    public void setFid(int fid) {
        this.fid = fid;
    }
    public String getFname() {
        return fname;
    }
    public void setFname(String fname) {
        this.fname = fname;
    }
    public byte[] getFile() {
        return file;
    }
    public void setFile(byte[] file) {
        this.file = file;
    }
    public String getIpaddress() {
        return ipaddress;
    }
    public void setIpaddress(String ipaddress) {
        this.ipaddress = ipaddress;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getCard() {
        return card;
    }
```

```
}  
public void setCard(String card) {  
    this.card = card;  
}  
public int getUid() {  
    return uid;  
}  
public void setUid(int uid) {  
    this.uid = uid;  
}  
public String getUtype() {  
    return utype;  
}  
public void setUtype(String utype) {  
    this.utype = utype;  
}  
public String getUname() {  
    return uname;  
}  
public void setUname(String uname) {  
    this.uname = uname;  
}  
public String getPassword() {  
    return password;  
}  
public void setPassword(String password) {  
    this.password = password;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}
```

```

public String getMobile() {
    return mobile;
}
public void setMobile(String mobile) {
    this.mobile = mobile;
}
public String getDate() {
    return date;
}
public void setDate(String date) {
    this.date = date;
}
}

```

### **Index.jsp**

```

<!DOCTYPE html>
<html>
<title>welcome</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body,h1 {font-family: "Montserrat", sans-serif}
img {margin-bottom: -7px}
.w3-row-padding img {margin-bottom: 12px}
</style>

<body style="background-image: url('images/cloud security.jpg'); background-size:
cover; background-attachment: fixed;">

<!-- SideBar -->
<jsp:include page="menu.jsp"></jsp:include>
<!-- !PAGE CONTENT! -->

```

```

<div class="w3-content" style="max-width:1500px;" >
<!-- Header -->
<div class="w3-opacity">
<span class="w3-button w3-xxlarge w3-white w3-right" onclick="w3_open()"><i
class="fa fa-bars"></i></span>
<div class="w3-clear"></div>
<header class="w3-center w3-margin-bottom">
  <h1><b style="color: white;font-size: 140%;">HoneyPot Network for Cloud
Security</b></h1>
</header>
</div>
</div>
<script>
// Toggle grid padding
function myFunction() {
  var x = document.getElementById("myGrid");
  if (x.className === "w3-row") {
    x.className = "w3-row-padding";
  } else {
    x.className = x.className.replace("w3-row-padding", "w3-row");
  }
}
// Open and close sidebar
function w3_open() {
  document.getElementById("mySidebar").style.width = "100%";
  document.getElementById("mySidebar").style.display = "block";
}
function w3_close() {
  document.getElementById("mySidebar").style.display = "none";
}
</script>
</body>
</html>

```

## **5. SCREENSHOTS**

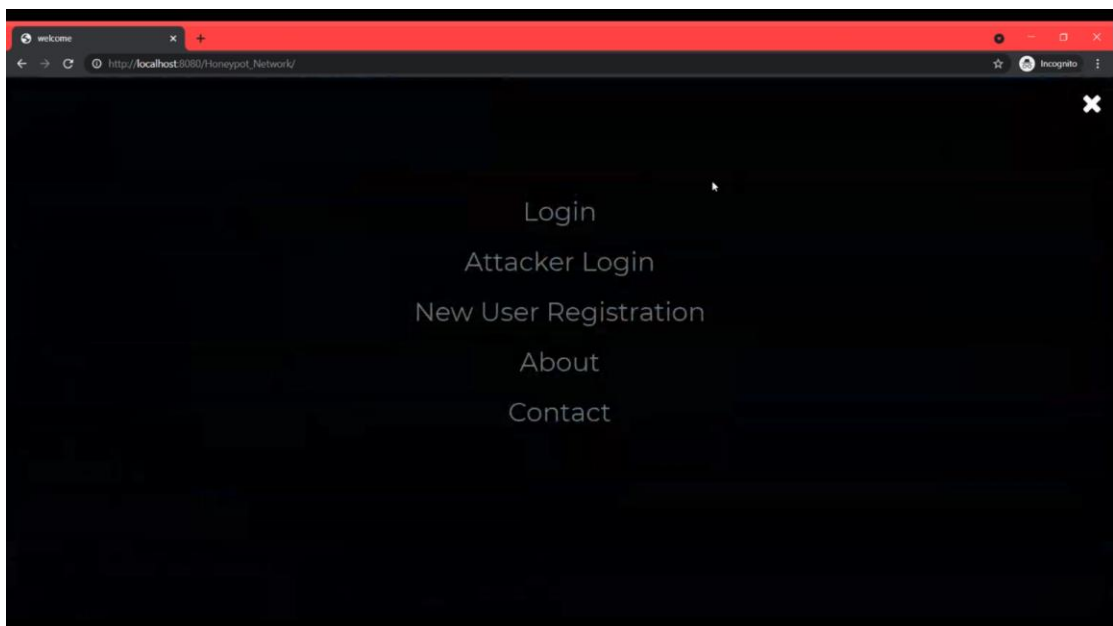
## 5. SCREENSHOTS

### 5.1 HOME PAGE



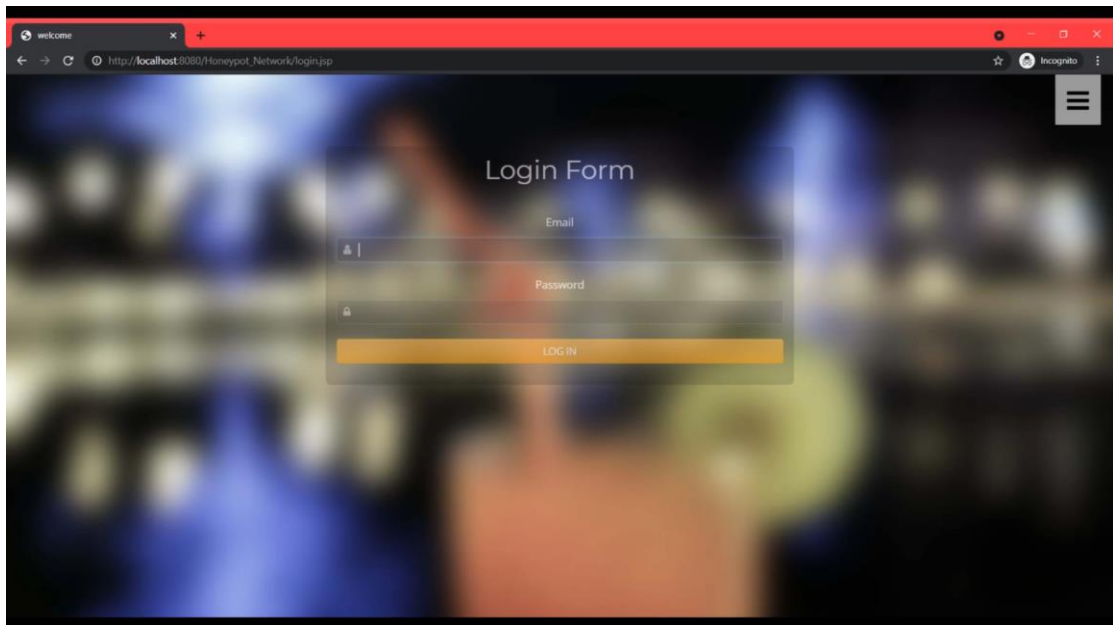
Screenshot 5.1: Home Page

#### 5.1.1 HOME PAGE MENU



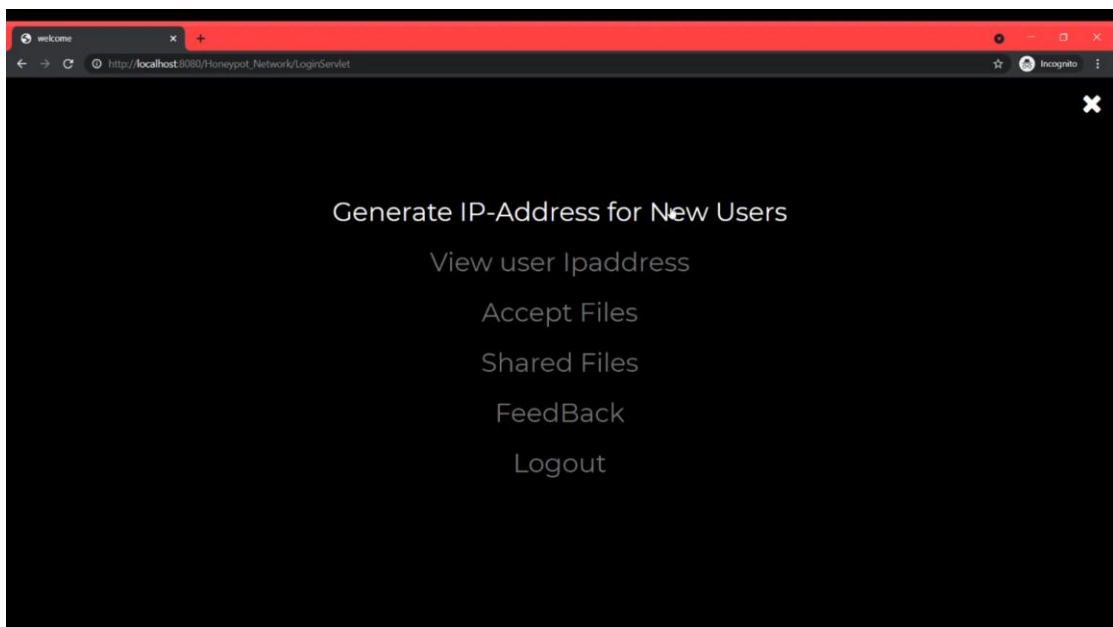
Screenshot 5.2: Home Page Menu

## 5.2 LOGIN PAGE



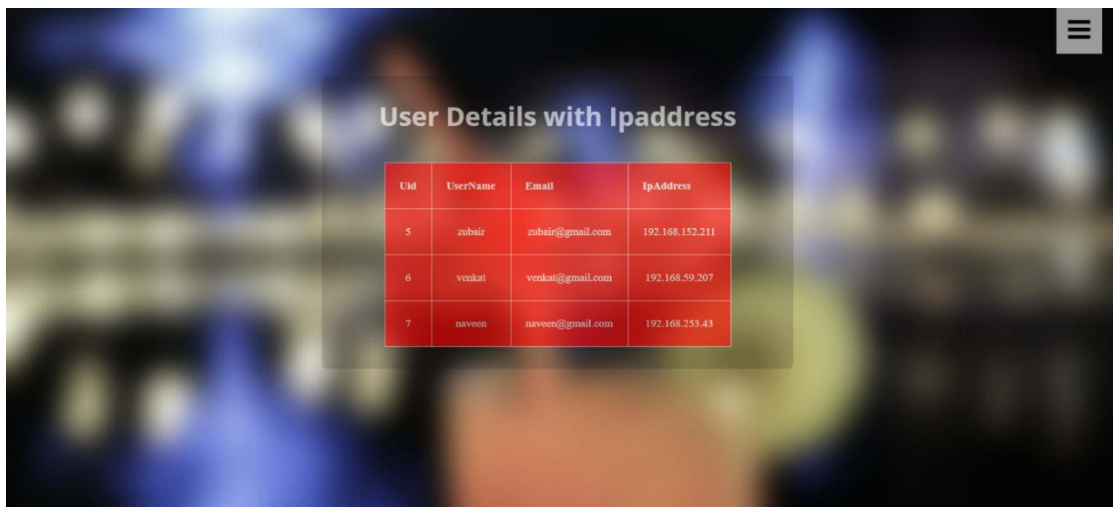
Screenshot 5.3: Login Page

## 5.3 ADMIN LOGIN MENU



Screenshot 5.3: Admin Login Menu

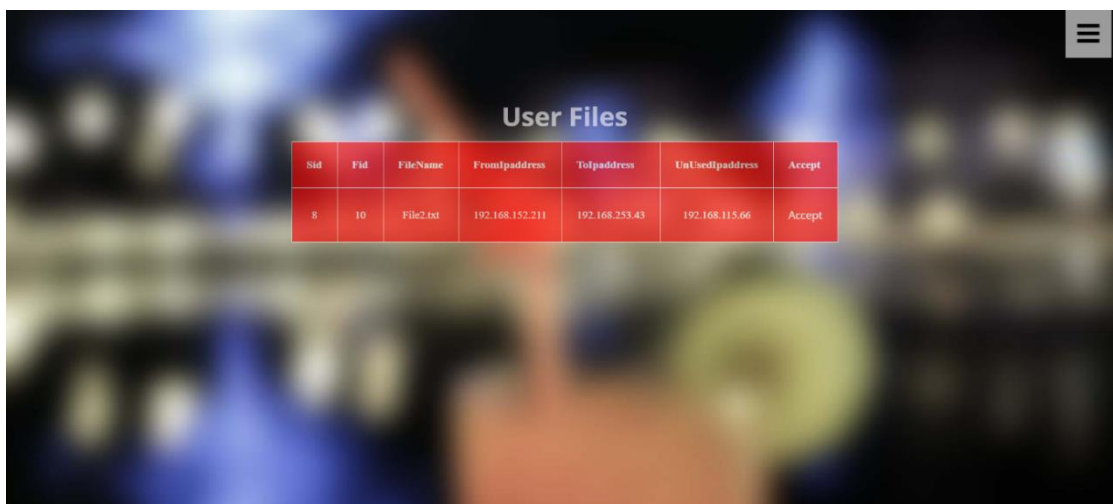
### 5.3.1 USER DETAILS



Uid	UserName	Email	IpAddress
5	zubair	zubair@gmail.com	192.168.152.211
6	venkat	venkat@gmail.com	192.168.59.207
7	naveen	naveen@gmail.com	192.168.253.43

Screenshot 5.3.1: User Details with IP address

### 5.3.2 ACCEPTANCE PAGE FOR FILES



Sid	Fid	FileName	FromIpaddress	ToIpaddress	UnUsedIpaddress	Accept
8	10	File2.txt	192.168.152.211	192.168.253.43	192.168.115.66	Accept

Screenshot 5.3.2: Admin verifies and accepts the files to be shared



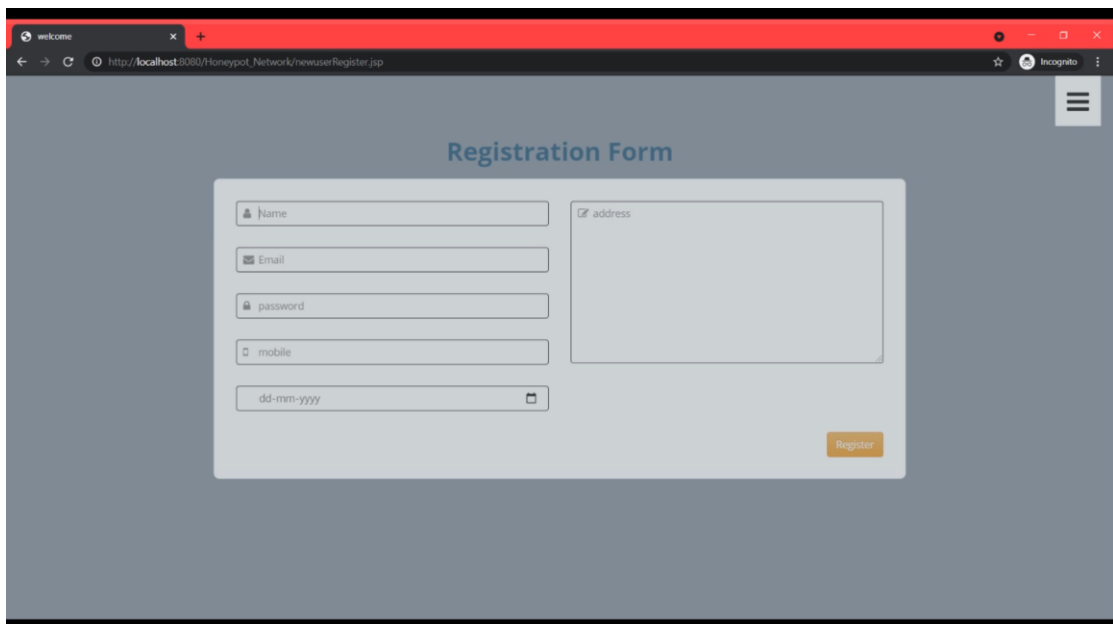
### 5.3.3 LIST OF THE FILES SHARED BY THE USERS



Sid	Fid	FileName	FromIpaddress	ToIpaddress	UsedIpaddress
1	2	hai.txt	192.168.218.134	192.168.63.226	192.168.174.197
2	3	D:\subir.txt	192.168.23.30	192.168.63.226	192.168.112.200
3	4	Sample.txt	192.168.87.202	192.168.78.109	192.168.221.63
4	6	Important files.txt	192.168.199.2	192.168.19.231	192.168.1.235
5	7	Important files.txt	192.168.59.207	192.168.152.211	192.168.17.215
6	8	Updated file.txt	192.168.152.211	192.168.59.207	192.168.41.48
7	9	File1.txt	192.168.253.43	192.168.152.211	192.168.118.1

Screenshot 5.3.3: List of the files shared between users

### 5.4 REGISTRATION FORM

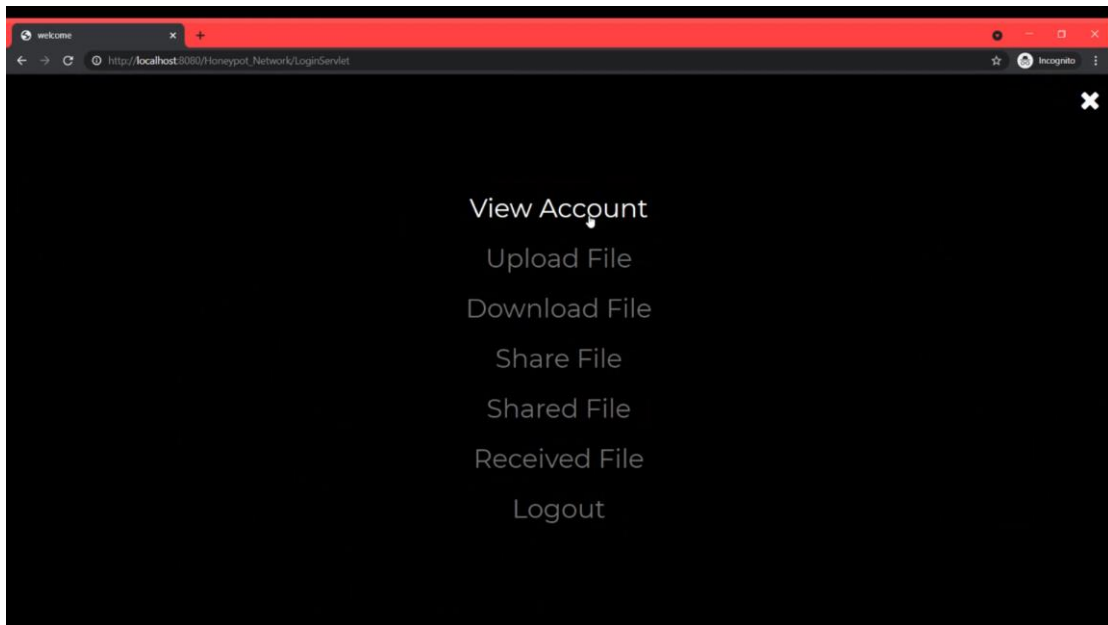


The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/Honeypot_Network/newuserRegister.jsp`. The page title is 'welcome'. The main content area is titled 'Registration Form' and contains a form with the following fields:

- Name (text input)
- Email (text input)
- password (password input)
- mobile (text input)
- dd-mm-yyyy (date input)
- address (text area)
- Register (button)

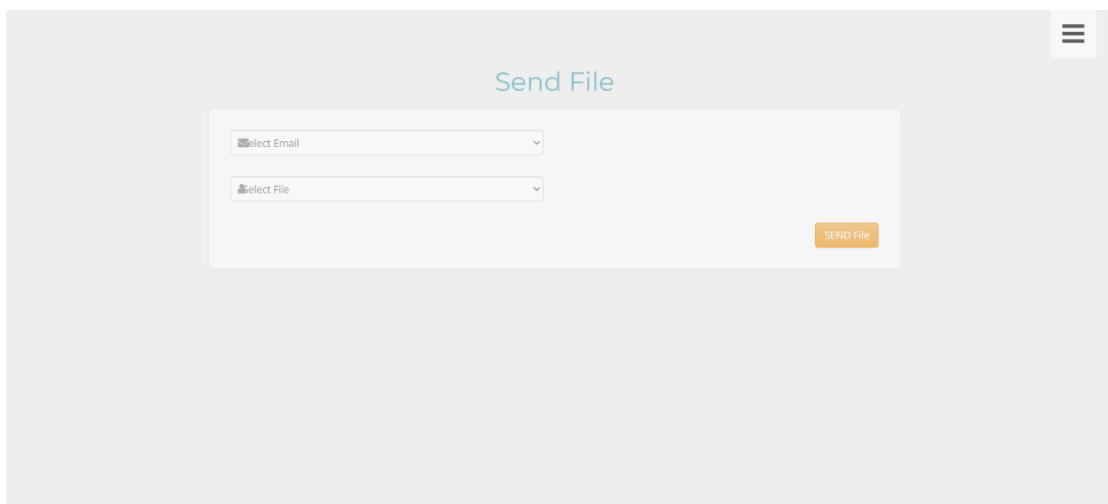
Screenshot 5.4: Registration Form

## 5.5 USER LOGIN MENU



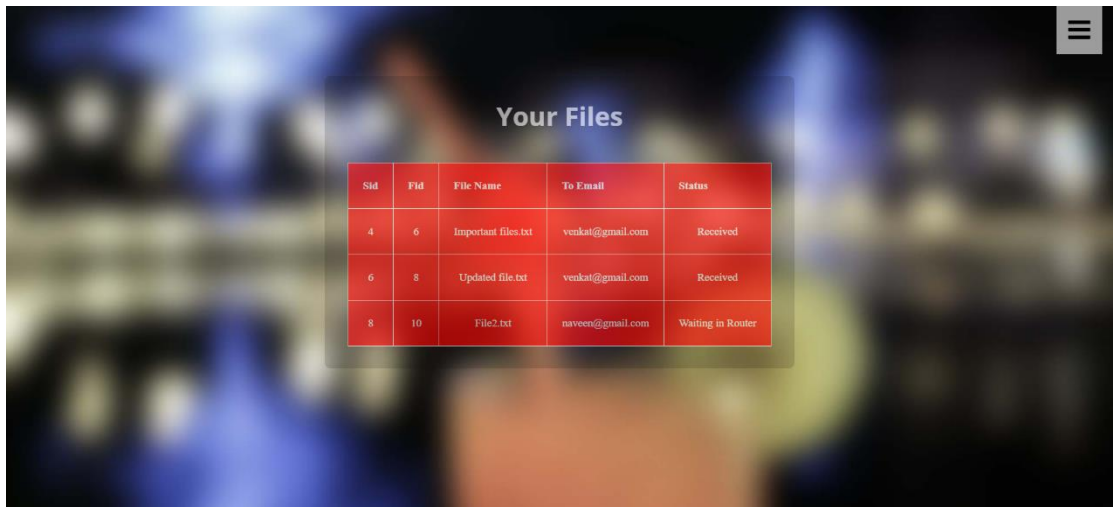
Screenshot 5.5: User Login Menu

### 5.5.1 USER SENDS FILES



Screenshot 5.5.1: User selects the files to be sent

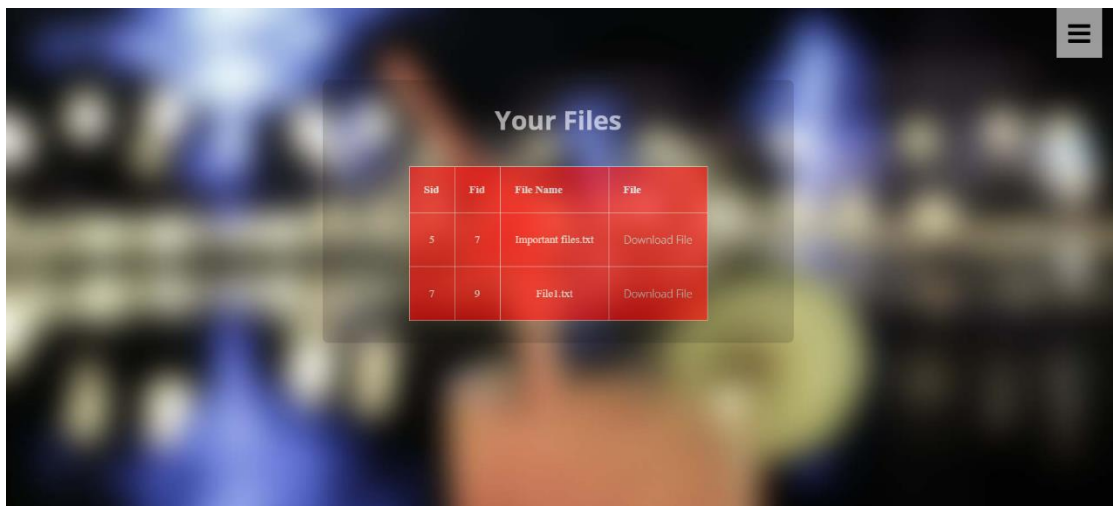
### 5.5.2 FILES SHARED BY USER



Sid	Fid	File Name	To Email	Status
4	6	Important files.txt	venkat@gmail.com	Received
6	8	Updated file.txt	venkat@gmail.com	Received
8	10	File2.txt	naveen@gmail.com	Waiting in Router

Screenshot 5.5.2: List of Files shared by Users

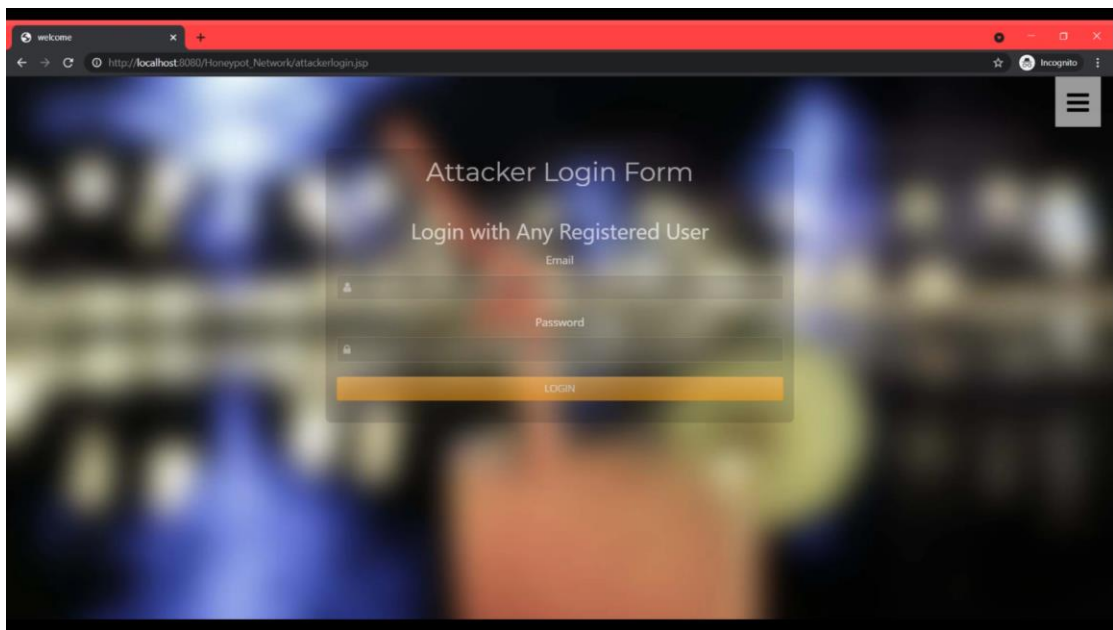
### 5.5.3 FILES RECEIVED BY USER



Sid	Fid	File Name	File
5	7	Important files.txt	Download File
7	9	File1.txt	Download File

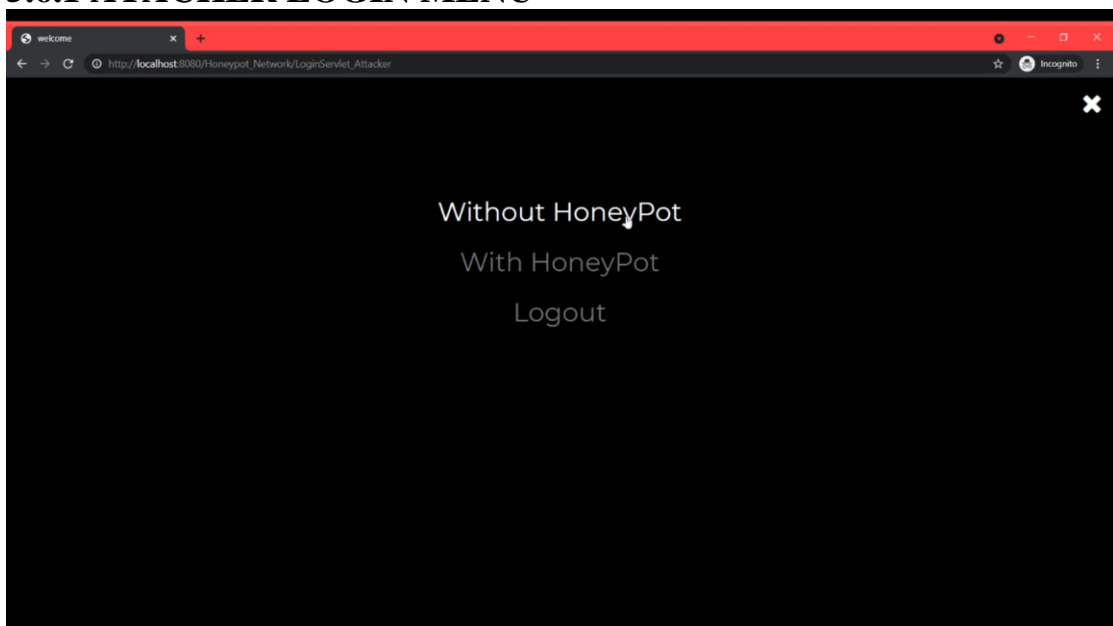
Screenshot 5.5.3: List of Files received by the user

## 5.6 ATTACKER LOGIN FORM



Screenshot 5.6: Attacker Login Form

### 5.6.1 ATTACKER LOGIN MENU



Screenshot 5.6.1: Attacker Login Menu

## **6. TESTING**

## **6. TESTING**

### **6.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **6.2 TYPES OF TESTING**

#### **6.2.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **6.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

## 6.3 TEST CASES

### 6.3.1 TEST CASES PASSED

S.No	Test case Description	Actual value	Expected value	Result
1	Create new user registration process	Enter the personal info and address info.	Insert personal info and address info in to oracle database successfully	True
2	Enter the username and password	Verification of login details.	Login Successfully	True
3	User File Share	Select User and select file and send Successful	Send successful	True
4	Router Accept File and send to Receive	Accepted Successful	Accepted and update data base	True
5	Attacker using unused Ippaddress and attack file	Attacked Successful	Show attacked file to attacker	True

**6.3.2 TEST CASES FAILED**

<b>S.No</b>	<b>Test case Description</b>	<b>Actual value</b>	<b>Expected value</b>	<b>Result</b>
<b>1</b>	Create the new user registration process	Enter the personal info and address info.	Personal info and address info its not update into database successfully.	False
<b>2</b>	Enter the username and password	Verification of login details if fails.	Login Success	False
<b>3</b>	User File Share	Select User and select file and send but it fails	Send successful	False
<b>4</b>	Router Accept File and send to Receive	Not Accepted	Accepted and update data base	False
<b>5</b>	Attacker using unused Ippaddress and attack file	Failed to Attack and file not shown	Show attacked file to attacker	False



## **7. CONCLUSION**

## **7. CONCLUSION AND FUTURE SCOPE**

### **7.1 PROJECT CONCLUSION**

Any Organization or firm with either outside resources/areas or cloud administrations ought to deploy cloud-based Honeypots because Security matters a lot. The IT staff might be required to arrange the Honeypots, yet the genuine outline ought to be driven by the security groups will's identity observing for vindictive movement or attacker. Any association managing delicate information in the cloud must prefer Honeypots, and they will likewise require talented system heads to screen the logs and respond to the information.

### **7.2 FUTURE SCOPE**

Cloud-based Honeypots give the capacity to explore and examine assaults that hit ordinary customers. Cloud is one of the few technologies that can bring about a major change, hence it is very necessary to make the security of the cloud stronger. In this paper, we present a way to tackle malicious users using Honeypot. Having them permits a specialist to interpret the IP locations and malware being utilized into security content that can ensure a normal cloud environment. Once those IP addresses have been distinguished, they will then lead a ping scope and defenselessness output to discover a shortcoming in the system outline or vulnerabilities in programming that can be misused.

## **8. BIBLIOGRAPHY**

## 8. BIBLIOGRAPHY

### 8.1 GitHub:

<https://github.com/ZAE3600/Honeypot-for-Cloud-Security>

### 8.2 REFERENCES

1. Cay S. Horton and Gary Cornell, “Core Java™ 2 Volume I – Fundamentals 7<sup>th</sup> Edition Pearson Education – Sun Microsystems, 17<sup>th</sup> August 2000.
2. Thau “The Book of JavaScript 2<sup>nd</sup> Edition” SPD, Dec 06, 2006
3. George Reese “Java Database Best Practices” O’Reilly- SPD, May – 2003.
4. Norman Richards and Sam Griffith “JBoss – A Developer Notebook” O’Reilly- SPD - June 2005
5. Herbert Shield “Java Complete Reference” 11<sup>th</sup> Edition, Dec 14, 2018

### 8.3 WEBSITES

1. <https://docs.oracle.com/javase/8/>
2. <https://www.javatpoint.com/servlet-tutorial>
3. <https://www.baeldung.com/intro-to-servlets>
4. <https://www.google.com/>

# HONEYPOT FOR CLOUD SECURITY

MHD. Zubair Hussain<sup>1</sup>, Ch. Naveen<sup>2</sup>, J. Venkat<sup>3</sup>, Kavitha Rani Balmuri<sup>4</sup>, Srinivas Konda<sup>5</sup>

<sup>1,2,3</sup>Student, <sup>4,5</sup>Professor

Department of CSE & IT, CMR Technical Campus, Hyderabad

## Abstract:

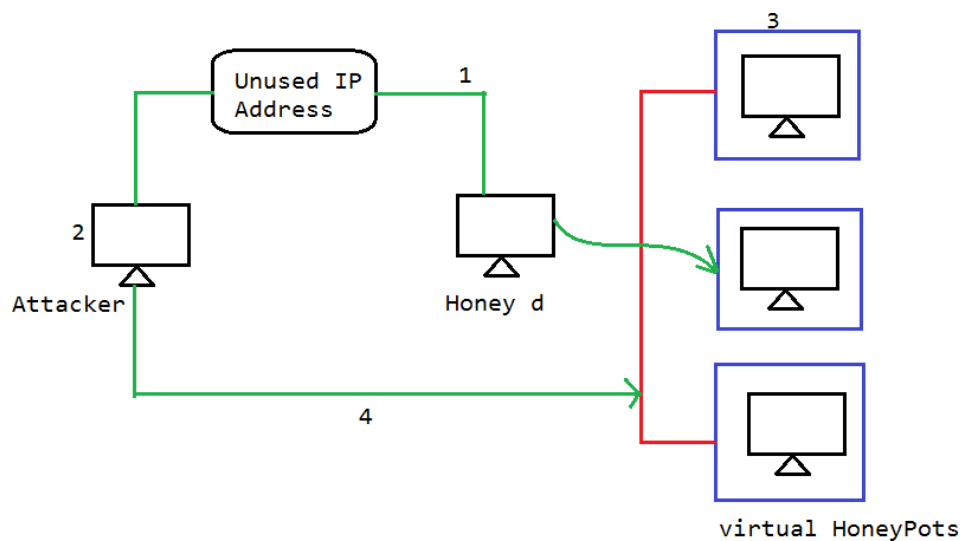
The World is getting more modest on account of innovation. With the quick expansion in the number of clients, there is an ascent in issues identified with equipment disappointment, web facilitating, space and memory distribution of information, which is straightforwardly or in a roundabout way, prompting information deficiency. We go to distributed computing rehearses with the target of offering types of assistance that are solid, quick, and low in cost. With enormous improvement in this innovation, there is a consistently expanding possibility of its security being undermined by noxious clients. An approach to redirect malignant traffic from frameworks is by utilizing Honeypot. The upsides of a low-connection honeypot are their effortlessness. These honeypots will in general be simpler to send and keep up, with negligible danger. For the most part, they include introducing programming, choosing the working frameworks and administrations you need to imitate and screen, and releasing the honeypot from that point. It is a gigantic methodology that has given indications of progress in the security of frameworks. Remembering the different legitimate issues, one may confront while sending Honeypot on outsider cloud seller workers, the idea of Honeypot is executed in a document sharing application that is conveyed on a cloud worker. This talks about the recognition assaults in a cloud-based climate just as the utilization of Honeypot for its security, along these lines proposing another method to do likewise.

## I. INTRODUCTION

Cloud computing is a strategy to store, offer and access information whenever and anyplace with a gadget that is associated with an organization, ideally the web. Distributed computing comprises of expandable extra room with no actual extra room which is available from anyplace on the planet utilizing any gadget, by interfacing it to the web. It contains an enormous number of registering gadgets associated through constant correspondence (the web) and has typical information stockpiling territory. The expression "the cloud" is utilized as a similitude for the Internet, in light of the way that a cloud-like shape was utilized to demonstrate network phone schematics, and later the Internet as a reflection of the hidden framework it addresses

Honeypots are seen as an effective strategy to follow software engineer's directions and inspire the suitability of safety instruments. Honeypots are explicitly intended to intentionally draw in and hoodwink programmers as well as recognize malevolent exercises performed over the Internet and can be considered a powerful strategy to follow programmer conduct. Honeypots can be characterized as frameworks or resources which are utilized to trap, screen yet, in addition, distinguish incorrect solicitations present inside an organization. They fluctuate in the connection gave to the aggressors, from low collaboration to medium and high, each type enjoys its benefits and burdens. They intend to investigate, get, watch and track the aggressor's conduct to make frameworks that are secure as well as handle such traffic. It is a firmly observed figuring asset that we need to be tested, assaulted, or bargained. "All the more definitely, it is a data framework asset whose worth lies in unapproved or illegal utilization of that asset".

Our extension is that Cloud-based Honeypots enable us to investigate and look at attacks that hit conventional clients. Having them allows an expert to decipher the IP areas and malware being used in security content that can guarantee a typical cloud climate. When those IP addresses have been recognized, they will at that point lead a ping extension and helplessness yield to find a deficiency in the framework blueprint or weaknesses in programming that can be abused.



## II. LITERATURE SURVEY

**A Review of Cloud Computing Security Issues:** Cloud computing is an arising worldview that has become the present most sizzling exploration region because of its capacity to decrease the expenses related to figuring. In the present time, it is the most intriguing and tempting innovation which is offering administrations to its clients on-request ridiculous. Since Cloud registering stores the information and its dispersed

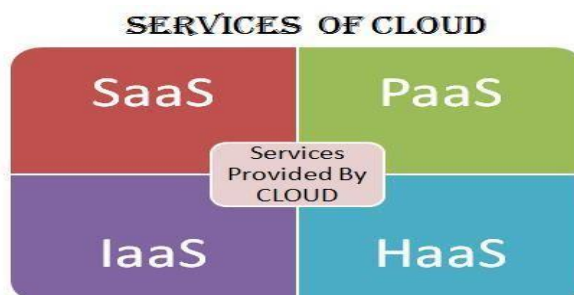
assets in the climate, security has become the fundamental impediment that is hampering the organization of cloud conditions. A few clients utilized the cloud to store their own information with the goal that information stockpiling security is needed on the capacity media. The significant worry of the cloud climate is security during transfer the of information on a cloud worker. Information stockpiling at cloud workers pulled in a mind-blowing measure of thought or spotlight from various networks. For rethinking the information there is a requirement for an outsider. The significance of an outsider is to forestall and control unapproved admittance to the information store in the cloud. This exploration paper examines the security issues of distributed storage.

The expression "Distributed computing" is the figuring administrations in Information Technology like framework, stages, or applications that could be organized and utilized through the web. The framework whereupon the cloud is constructed is a huge scaled conveyed foundation wherein a common pool of assets is by and large virtualized, and administrations that are offered are circulated to customers as far as virtual machines, arrangement climate, or programming. Thus it very well may be handily reasoned that as per the prerequisites and current jobs, the administrations of the cloud could be scaled progressively. As numerous assets are utilized, they are estimated, and afterward, the installment is made dependent on the utilization of those assets.

As per the definition of [15], distributed computing is " It is a huge dispersed registering model that is coordinated by monetary reasonability of equilibrium, in which stake of detaching, key, stacking, platform in which offices are provided according to the solicitation of outside unfamiliar customers through the web". There are a few instances of cloud administrations like webmail, online document, and business applications. Distributed computing gives a common pool of assets, including information extra room, organizations, PC handling power, and concentrated corporate and client applications. Distributed storage [20] determines the capacity on the cloud with practically economical capacity and reinforcement alternative for little endeavors. The real stockpiling area might be in a solitary stockpiling climate or imitated to numerous worker stockpiling dependent on the significance of the information. The component [20] model of distributed storage comprises four layers: stockpiling layer which stores the information, fundamental administration layer which guarantees security and solidness of distributed storage itself, application interface layer which gives application administration stage, and access layer which gives the entrance stage.

**Cloud Infrastructure as a service(IaaS):** In this composition of implemented environment for their system a supplier must be supply a different computing resources which include loading, processing unit. Client has flexile to achieve and switches software mutilated to be implemented and vary between different applications like operating system etc.

**Cloud Platform as a service (PaaS):** This software supplies client with the ability to establish and extended applications that are mainly positioned on equipment and programming languages promoted by the suppliers. In this the client has no containment over the different organization but has containment over the extended 34 A Review of Cloud Computing Security Issues applications. Examples of this class of services include Google App Engine, Windows Azure Platform and rack space.



### **Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures:**

Web and organizations application is developing quick, so the need to secure such application is expanded by utilizing cryptographic strategies. The two generally acknowledged and utilized cryptographic strategies are symmetric and topsy-turvy. The DES in a perfect world has a place with the classification of symmetric key cryptography and RSA has a place with the classification of topsy-turvy key cryptography. This paper involves a short portrayal of RSA and DES cryptography calculations and their current weaknesses alongside their countermeasures. Other than this, there is a hypothetical exhibition investigation and correlations of symmetric and lopsided cryptography

Numerous encryption calculations are generally accessible and utilized in data security. They can be ordered into Symmetric (private) and Asymmetric (public) keys encryption. In Symmetric keys encryption or mystery key encryption, just one key is utilized to encode and unscramble information. The key ought to be appropriated before transmission between substances. Keys assume a significant part. In the event that the powerless key is utilized in the calculation, everybody may unscramble the information. The strength of Symmetric-key encryption relies upon the size of the key utilized. For a similar calculation, encryption utilizing a more extended key is harder to break than the one done utilizing a more modest key. There are numerous instances of solid and feeble keys of cryptography calculations like RC2, DES, 3DES, RC6, Blowfish, and AES. RC2 utilizes one 64-digit key. DES utilizes one 64-bits key. Triple-DES (3DES) utilizes three 64- bits keys while AES utilizes different (128,192,256) bits keys. Blowfish utilizes different (32-448); default 128bits while RC6 is utilized different (128,192,256) bits keys [1-4]. In any case, the fundamental issue with this is the protected transmission of key over the vindictive organization. Topsy-turvy key encryption or public-key encryption is utilized to tackle the issue of key dispersion. In Asymmetric keys, two keys are utilized; private and public keys.



The public key is utilized for encryption and the private key is utilized for unscrambling.

(For example RSA and Digital Signatures). Since clients will in general utilize two keys: public key, which is known to the general population, and private key which is known uniquely to the client. There is no requirement for dispersing them preceding transmission. Nonetheless, public-key encryption depends on numerical capacities, computationally serious, and isn't extremely proficient for little cell phones.

Lopsided encryption procedures are right around multiple times more slow than Symmetric strategies since they require more computational handling power. The most well-known characterization of encryption methods.

### **Ensuring Data Storage Security in Cloud Computing:**

Cloud computing is another computational worldview that offers an imaginative plan of action for associations to receive IT without forthright ventures. In spite of the potential addition accomplished from distributed computing. It is plainly one of the present most tempting innovation zones due, in any event to some extent, to its expense effectiveness and adaptability. Distributed computing moves the application programming and information base to the enormous server farm where the information the executives and administrations may not be completely reliable. In this article our significant conversation on cloud information stockpiling security. Security is a significant part of the nature of administrations. To guarantees the rightness of client information in the cloud. We propose a successful and adaptable dispersed plan of two-way handshakes dependent on symbolic administration. By using the homomorphic token with a conveyed check of deletion coded information, our plan accomplishes the incorporation of capacity rightness protection and information mistake limitation i.e., the distinguishing proof of acting up a server(s).

Today, the fourteenth biggest programming organization by market capitalization (salesforce.com) works predominantly in the cloud, the best five programming organizations by deals income all have significant cloud contributions, and the market, in general, is anticipated to develop to \$160 Billion by 2011 (source: Merrill Lynch). However in spite of the trumpeted business and specialized benefits of distributed computing, numerous potential cloud clients presently can't seem to join the cloud, and those significant partnerships that are cloud clients are generally placing just their less delicate information in a cloud. The absence of control in the cloud is a significant concern. One part of control is straightforwardness in the cloud execution – to some degree in spite of the first guarantee of distributed computing wherein the cloud execution isn't significant. Straightforwardness is required for guidelines.

Distributed computing addresses a new change in outlook for the arrangement of processing foundation which rethinks calculation and capacity prerequisites of

utilizations and administrations to an oversight framework. Distributed computing definitely presents new testing security dangers for various reasons.

➤ Cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data in cloud computing. The problem of verifying correctness of data storage in cloud is becoming even more challenging.

➤ Cloud computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, like deleting, modification, insertion, recording, etc. To ensure storage correctness under dynamic data update, this dynamic feature also makes traditional integrity insurance technique futile and entails new solutions.

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization on data errors, i.e., the identification of the misbehaving server(s).

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats.

1. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

2. Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance.

3. Third is the deployment of cloud computing, it is powered by data centers running in a simultaneous cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats.

### **Ensuring Data Storage Security in Cloud Computing**

Cloud Computing has been envisioned as the next generation architecture of IT

Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazon's S3 is such an example.

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection can not be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under

dynamic data update is hence of paramount importance. However, this dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world. However, such important area remains to be fully explored in the literature.

Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works. These techniques, while can be useful to ensure the storage correctness without having users possessing data, can not address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

Our work is among the first few ones in this field to consider distributed data storage in Cloud Computing. Our contribution can be summarized as the following three aspects:

1. Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.
2. Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
3. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

### **Modified Caesar Cipher for Better Security Enhancement**

Encryption is the process of scrambling a message so that only the intended recipient

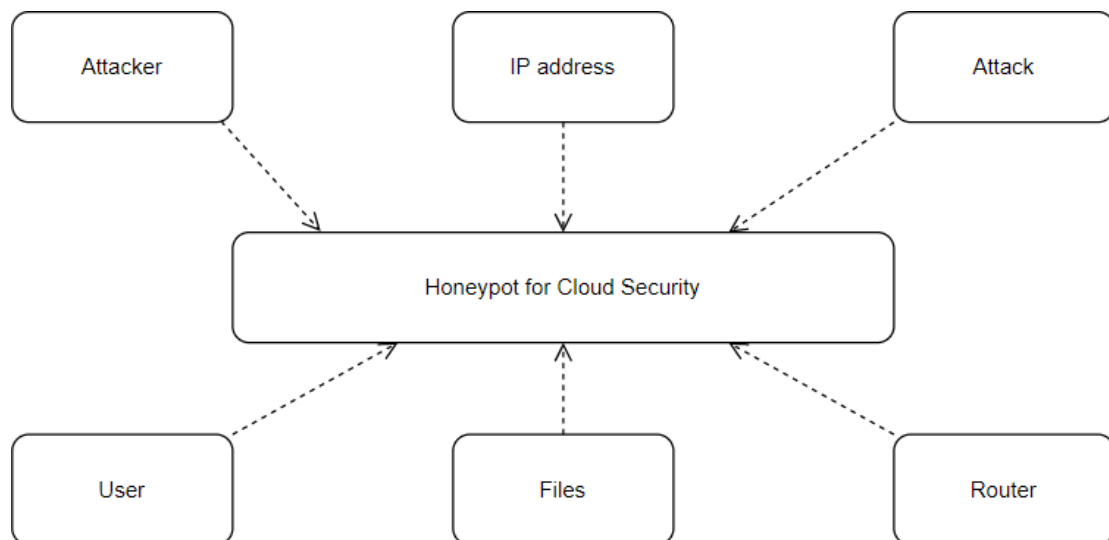
can read it. With the fast progression of digital data exchange in electronic way, Information Security is becoming much more important in data storage and transmission. Caesar cipher is a mono alphabetic cipher. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter. In this paper, author modified the traditional Caesar cipher and fixed the key size as one. Another thing alphabet index is checked if the alphabet index is even then increase the value by one else alphabet index is odd decrease the key value by one. Encryption and scrambling of the letters in the Cipher Text.

Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form nonrecognizable by its attackers while stored and transmitted. Security is a big concern and securing crucial data is very essential, so that the data cannot be change or misused for any illegal purposes. For example in Internet Banking system, e-reservation system the security of data is a very important issue. Under no circumstances the intruder should be able to get into the server database or the confidential data. In any type of service sectors the confidentiality of data is a very important issue. The primary goal of any system is that the data cannot be modified by any external user or intruder. To avoid such a type of situation. Convert data into a non readable form at sender side and convert that data in readable form again at receiver side. The technique and science of creating non readable data or cipher so that only authorized person is only able to read the data is called Cryptography. In Cryptography, Caesar cipher is one of the most widely known encryption decryption algorithm. Caesar cipher is a type of substitution type cipher in this kind of cipher each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. The encryption is represented using modular arithmetic.

With the increasing trend of internet technologies, numerous security issues are arising. Cloud users are also victim of the security issues. In cloud computing security issues are faced by the Cloud providers as well customers. In most cases, provider must ensure that their infrastructure is secure and that their client's personal data and applications are protected while the customer must ensure that the Cloud provider has taken the proper security measures to protect their information So security issues are everywhere.

### **III Proposed Methodology:**

This paper proposes another strategy of securing information and assets in a cloud through Honeypot by carrying out it through an application on the previously mentioned foundation (Cloud Computing Environment). Numerous limitations should be followed while executing a honeypot. The application makes it conceivable to store just as offer a report. While sharing or transferring the record it is encoded utilizing a secret key. Assuming the right secret word isn't given, no message would be shown rather the aggressor would be shown a vacant record. Since the genuine working of a Honeypot includes quiet discovery, subsequently the application tracks the IP address of the client so that later the administrator can audit it and perceive the malevolent element.



#### IV Conclusion:

Any association or firm with either outside assets/territories or cloud organizations should send cloud-based Honeypots. The IT staff may be needed to orchestrate the Honeypots, yet the real layout should be driven by the security gatherings will's character noticing for pernicious development. Any affiliation overseeing sensitive data in the cloud should lean toward Honeypots, and they will in like manner require skilled framework heads to screen the logs and react to the data.

#### References:

- [1] Cay S. Hortman and Gary Cornell, "Core Java™ 2 Volume I – Fundamentals 7<sup>th</sup> Edition Pearson Education – Sun Microsystems, 17<sup>th</sup> August 2000.
- [2] Cay S. Hortman and Gary Cornell, "Core Java™ 2 Volume II – Advanced Pearson Education – Sun Microsystems, Aug 17, 2004.
- [3] Thau "The Book of JavaScript 2<sup>nd</sup> Edition" SPD, Dec 06, 2006.
- [4] Joshua Bloch "Effective Java-Programming Language Guide" Pearson Education-Sun Microsystems, May 08, 2008.
- [5] George Reese "Java Database Best Practices" O'Reilly- SPD, May – 2003.
- [6] Norman Richards and Sam Griffith "JBoss – A Developer Notebook" O'Reilly- SPD

- [7] Herbert Shield “Java Complete Reference”





**JASC**  
**JOURNAL OF APPLIED SCIENCE AND COMPUTATIONS**

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 1076-5131 / web : <http://j-asc.com/> / e-mail : [submitjasc@gmail.com](mailto:submitjasc@gmail.com)

Address : H.NO: C-72, Gali No: 3, Hardev Nagar, Jharoda, Burari, New Delhi - 110084

**CERTIFICATE OF PUBLICATION**

This is to certify that the paper entitled

**"HONEYPOT FOR CLOUD SECURITY"**

Authored by

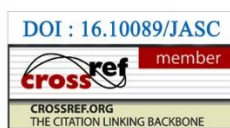
**MHD. Zubair Hussain**

From

**CMR Technical Campus, Hyderabad**

Has been published in

**JASC JOURNAL, VOLUME VIII, ISSUE V, MAY- 2021**



*N. Balasubramanian*  
Dr. N. BALASUBRAMANIAN  
Editor-In-Chief  
JASC  
<http://j-asc.com/>



**JASC**  
**JOURNAL OF APPLIED SCIENCE AND COMPUTATIONS**

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 1076-5131 / web : <http://j-asc.com/> / e-mail : [submitjasc@gmail.com](mailto:submitjasc@gmail.com)

Address : H.NO: C-72, Gali No: 3, Hardev Nagar, Jharoda, Burari, New Delhi - 110084

**CERTIFICATE OF PUBLICATION**

This is to certify that the paper entitled

**"HONEYPOT FOR CLOUD SECURITY"**

Authored by

**J. Venkat**

From

**CMR Technical Campus, Hyderabad**

Has been published in

**JASC JOURNAL, VOLUME VIII, ISSUE V, MAY- 2021**



*N. Balasubramanian*  
Dr. N. BALASUBRAMANIAN  
Editor-In-Chief  
JASC  
<http://j-asc.com/>







**JASC**  
**JOURNAL OF APPLIED SCIENCE AND COMPUTATIONS**

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 1076-5131 / web : <http://j-asc.com/> / e-mail : [submitjasc@gmail.com](mailto:submitjasc@gmail.com)

Address : H.NO: C-72, Gali No: 3, Hardev Nagar, Jharoda, Burari, New Delhi - 110084

**CERTIFICATE OF PUBLICATION**

This is to certify that the paper entitled

**“HONEYPOT FOR CLOUD SECURITY”**

Authored by

**Ch. Naveen**

From

**CMR Technical Campus, Hyderabad**

Has been published in

**JASC JOURNAL, VOLUME VIII, ISSUE V, MAY- 2021**



*N. Balasubramanian*  
Dr. N. BALASUBRAMANIAN  
Editor-In-Chief  
JASC  
<http://j-asc.com/>



**JASC**  
**JOURNAL OF APPLIED SCIENCE AND COMPUTATIONS**

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 1076-5131 / web : <http://j-asc.com/> / e-mail : [submitjasc@gmail.com](mailto:submitjasc@gmail.com)

Address : H.NO: C-72, Gali No: 3, Hardev Nagar, Jharoda, Burari, New Delhi - 110084

**CERTIFICATE OF PUBLICATION**

This is to certify that the paper entitled

**“HONEYPOT FOR CLOUD SECURITY”**

Authored by

**Kavitha Rani Balmuri, Professor**

From

**CMR Technical Campus, Hyderabad**

Has been published in

**JASC JOURNAL, VOLUME VIII, ISSUE V, MAY- 2021**



*N. Balasubramanian*  
Dr. N. BALASUBRAMANIAN  
Editor-In-Chief  
JASC  
<http://j-asc.com/>

